Double and Multiple-scattering effects in translucent materials

Nicolas Holzschuch and Jean-Dominique Gascuel
INRIA Grenoble Rhône-Alpes and LJK (UMR 5224), CNRS and Université de Grenoble

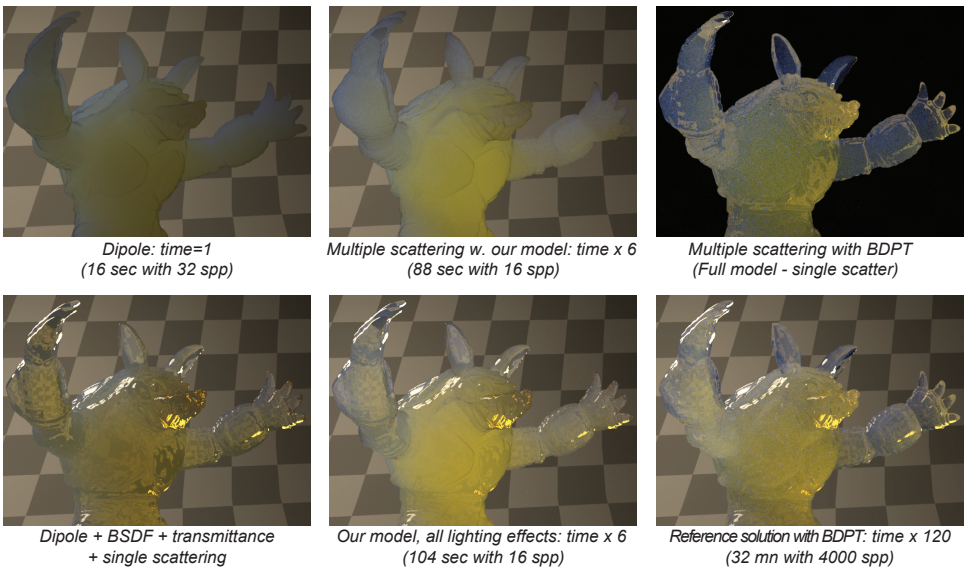


Figure 1: Comparison between our algorithm and standard algorithms for homogenenous scattering material[1]. Top row: multiple scattering effects only. Bottom row: full light simulation (reflection, refraction single and multiple scattering). The dipole approximation (left) results in the wrong color. Reference solution (right), computed using bi-directional path tracing (BDPT), takes a long time. Multiple scattering effects extrapolated from double scattering computed using our algorithm (center) is 20 times faster than the reference solution and provides the right color.

Abstract
Some materials, such as coffee, milk or marble, have a soft translucent aspect because of sub-surface scattering: light enters them, is scattered several times inside before leaving in a different place. A full representation of sub-surface scattering effects in illumination simulation is computationally expensive. The main difficulty comes from multiple scattering events: the high number of events increases the uncertainty on the result, forcing us to allocate more time for the computations. In this paper, we show that there is a strong correlation between the surface effects of multiple scattering inside the material and the effects after just two scatter events. This knowledge will help in accelerating multiple scattering effects. We exploit this knowledge to provide a model and implementation for fast computation of double-scattering events, using a precomputed density function stored in a compact way.

1. Material: $\sigma_t = (0.164, 0.26, 1.192)$, $\alpha = 0.95$, $\eta = 1.3$, $g = 0.3$.

1 Introduction

Many materials, such as milk or wax, exhibit *sub-surface scattering* behaviour: the light that reaches them enters inside, and is scattered several times before leaving again. This effect gives the material a slightly translucent look. The overall aspect depends on how far the light can penetrate inside the material, and on how many times it will be scattered before the exit. This effect is visible on solids such as marble and jade, as well as liquids such as milk, coffee or orange juice. It results in a softer look for objects made with these materials.

We have to model and render this *sub-surface scattering* effect for photorealistic rendering of these materials. But taking it into account greatly increases the computational complexity of illumination simulation. In order to compute the outgoing light at a specific point, we now have to take into account the incoming light at all neighbouring points, from all directions, instead of just the light incoming on this point. This adds two dimensions to the sampling, increasing the computation time. We also have to store the behaviour of the material, a function that express the relationship between incoming and outgoing light. Because the relationship is both spatial and angular, this function, the BSSRDF, has 6 dimensions. With regular sampling on all directions, storage is prohibitive.

A full representation of sub-surface scattering effects is expensive both in storage and computation time. To reduce the complexity, we use approximations, such as the diffusion dipole. For materials with a small mean-free path, light encounters a large number of scattering events before exiting the material. As a consequence, outgoing light is diffuse, removing the angular dependency and reducing the number of dimensions. Although is is practical and efficient, this approximation is not sufficient to represent the full range of sub-surface scattering materials. In a seminal study, Donner *et al.* [DLR+09], proved that materials can exhibit strong directional response, with complicated lobes. They also provided a detailed and compact representation for BSSRDF lobes in the angular dimension, but still sample regularly in the spatial dimension.

In this paper, we build upon the work of Donner *et al.* [DLR+09]. We focus on multiple scattering effects (light being scattered several times before leaving the material). We reproduced the experimental study of Donner *et al.* [DLR+09], but separate between paths with two scatter events and others. Our study shows a strong correlation between the light leaving after two scatter events and the light leaving after multiple scatter events, for a large range of materials. As the complexity of paths with two-scatter events is much lower than the complexity of multiple-scatter events, we can use this knowledge to speed up scattering computations. In a second contribution, we provide a fast, low-memory algorithm for computing two-events sub-surface scattering and extrapolate for multiple scattering.

Our paper is organised as follows: in the next section, we briefly review relevant work on sub-surface scattering. In section 3, we present our study, comparing two-scatter events and multiple scatter events. We show there is a strong correlation between two-scatter events and multiple-scatter events, except for strongly backward scattering materials. In section 4, we present our model for compact representation of two-events sub-surface scattering; the probability density of twice-scattered photons can be represented independently of the incoming direction. We then use this probability density to compute the outgoing light by integrating along the ray. Finally, in section 6, we conclude and present directions for future work.

2 Previous work

We can do illumination simulation with sub-surface scattering effects by tracing photons inside the material, taking into account every scatter event [Kaj86]. This technique is highly expensive in computation time, resulting in many research for faster representation of scattering effects.

Single scattering events are complicated. When the camera and viewer are assumed to be at an infinite distance, Hanrahan and Krueger [HK93] showed that there single-scatter events can be represented as a BRDF. In the general configuration, single scatter events are highly directional and can result in caustics inside the material [WZHB09]. Subsequent research has focused on the multiple scattering problem: light being scattered several times before leaving the material.

When there is a large number of scattering events, exiting light has lost all directional information. It can be modelled using a diffuse representation, the *dipole approximation*, introduced by Jensen *et al.* [JMLH01]. When there is a small number of scattering events, the outgoing light keeps directional information. We need a more complex model to represent it. D'Eon and Irving [DI11] greatly increased the accuracy of the dipole model, using a better approximation for the diffusion inside the material. Their approach still loses directional information, while we keep it. The difference could be important for anisotropic materials.

Donner *et al.* [DLR+09] conducted a very thorough study of these sub-surface scattering materials. They worked both on Monte-Carlo simulations and measurements from actual materials. They showed that the material response has a strong directional component and that the shape of the lobes can be complex. They also provided a model for the directional component of the material response, using elliptic coordinates. But they did not provide for the spatial response, and sampled in concentric circles around the point of impact. Their model is the most accurate to date, but it has a high memory cost (up to 250 MB for each material). It is also limited to a single lobe. This paper extends and completes their study. We also provide a model that is less accurate, but much more compact and can represent multiple lobes.

3 Analysis of subsurface scattering

3.1 Properties of translucent materials

As light enters a translucent material (see Figure 1), it is first refracted at the interface using Snell's law: the light beam changes direction, while staying in the plane defined by the incident light beam and the normal at the entry point. The angle of the refracted ray with the normal depends on the index of refraction of the materials:

$$\eta \sin \theta_i' = \sin \theta_i$$

Where η is the ratio of the index of refraction. If the first material is empty or almost empty (such as air), η is equal to the index of refraction of the second material.

After being refracted at the interface, light travels inside the translucent material. Here, it can encounter particles, resulting in either absorption or scattering. If it is scattered, light changes directions and continues travelling. As we are interested in the outside appearance of the material, we keep track of light until it reaches the surface again. The effect of a translucent material on light can be modelled using a few parameters:

- The *absorption coefficient*, σ_a express how the material absorbs light. It is the reciprocal of the average length travelled by a photon before being absorbed.

- Similarly, the *scattering coefficient*, σ_s express how often the rays are scattered inside the material. Both coefficients have units in m^{-1}.
- The intensity along a ray decreases with the distance travelled, d:

$$I(d) = I_0 \mathrm{e}^{-(\sigma_s+\sigma_a)d} \tag{1}$$

  We call $\ell = (\sigma_s + \sigma_a)^{-1}$ the *mean-free path*. It corresponds to the average length a photon can travel before getting scattered or absorbed. It is expressed in m. We call $\sigma_t = \sigma_s + \sigma_a = 1/\ell$ the transmission coefficient.
- When a ray is scattered inside the material, the *phase function* describes the angular distribution of light after scattering. In this work, we used the Henyey-Greenstein phase function, which has rotational symmetry and depends on the angle between the direction of the ray before and after scattering:

$$p(\theta, g) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{\frac{3}{2}}} \tag{2}$$

  The parameter g express the *anisotropy* of the phase function. For $g = 0$, the rays are scattered equally in all directions. For $0 < g < 1$, the rays are scattered mainly in the same direction as the incoming ray (*forward scattering*). For $-1 < g < 0$, the rays are scattered mainly in the reverse direction (*backward scattering*).
  The Henyey-Greenstein phase function was originally developed for radiations in the galaxy [HG41]. It can be used to approximate any phase function ϕ using the parameter g computed as:

$$g = \int_\Omega \cos\theta\phi(\theta)\,\mathrm{d}\omega \tag{3}$$

  We will use this important property in our work.
- The *albedo* $\alpha = \frac{\sigma_s}{\sigma_s+\sigma_a}$, express the relative importance of scattering and absorption. Materials with a high albedo ($\alpha \approx 1$) have mostly scattering effects, with little absorption. Conversely, in materials with a low albedo ($\alpha \approx 0$), absorption effects dominate. The albedo is a dimensionless quantity, between 0 and 1.

As pointed out before (*e.g.* [DLR+09]), the mean-free path gives the scale of the response for a given material. For two materials that differ only by ℓ, we can predict the response of the second material by scaling the response of the first material by the ratio of their mean-free-paths.

In the remainder of this paper, we parametrise using normalised coordinates. All spatial coordinates are divided by ℓ to give a dimensionless, generic parametrisation. The behaviour of translucent materials only depends on three parameters: α, g and η.

Since we're interested only in the outside appearance of translucent materials, we model their behaviour using a bi-directional sub-surface scattering reflectance distribution function S (BSSRDF). It defines the general transport of light between two points and directions as the ratio of outgoing radiance at point $\boldsymbol{x}_o$ in direction $\boldsymbol{\omega}_o$, $L_o(\boldsymbol{x}_o, \boldsymbol{\omega}_o)$ to the incident radiant flux $\Phi_i(\boldsymbol{x}_i, \boldsymbol{\omega}_i)$ at point $\boldsymbol{x}_i$ from direction $\boldsymbol{\omega}_i$:

$$S(\boldsymbol{x}_i, \boldsymbol{\omega}_i, \boldsymbol{x}_o, \boldsymbol{\omega}_o) = \frac{\mathrm{d}L_o(\boldsymbol{x}_o, \boldsymbol{\omega}_o)}{\mathrm{d}\Phi_i(\boldsymbol{x}_i, \boldsymbol{\omega}_i)} \tag{4}$$

3.2 The experimental framework

Our entire experimental setting follows closely Donner *et al.* [DLR+09], for easier comparison with their results. We restrict ourselves to spatially uniform, homogeneous materials,

whose boundary is an infinite plane. This reduces the number of dimensions as we only need the relative positions in space and angle. We express the spatial difference $\boldsymbol{x}_o - \boldsymbol{x}_i$ in polar coordinates: r and θ_s. We orient our axes so that the angular dependence of the BSSRDF depends only on three angles $(\theta_i, \theta_o, \phi_o)$. For a given material, the BSSRDF depends on five parameters: $S(r, \theta_s, \theta_i, \theta_o, \phi_o)$.

We compute the BSSRDF using Monte-Carlo simulation. We send photons along a collimated beam, hitting the translucent material with a direction $\boldsymbol{\omega}_i$ making angle θ_i with the surface normal. These photons are refracted as they enter the material, and propagate a distance d before being scattered or absorbed:

$$d = \frac{-\log \xi}{\sigma_s + \sigma_a} \tag{5}$$

Where ξ is a random variable in $[0, 1]$. This expression for d corresponds to the exponential falloff for illumination.

When a photon is scattered, we compute its new direction by importance-sampling the phase function [Jen01] and modulate its intensity by the albedo. If a photon hits the interface, we compute its reflected direction and modify its power by applying the Fresnel term.

To increase the speed of our simulations, we select sample points on the surface of the material. For each sample point $\boldsymbol{x}_o$, for each photon in the simulation, we compute the power Φ_e that this photon would contribute if it was scattered directly to this sample point and average the contributions of all the photons:

$$\Phi_e = \alpha \; p(\tilde{\boldsymbol{d}} \cdot \boldsymbol{\omega}_p, g) \; \mathrm{e}^{-\frac{\|\boldsymbol{d}\|}{\ell}} \; F_t(\tilde{\boldsymbol{d}}) \Phi_p \tag{6}$$

- Φ_p is the power of the photon,
- $\boldsymbol{d}$ is the vector joining the position of the photon and the sample point,
- $\tilde{\boldsymbol{d}} = \boldsymbol{d}/\|\boldsymbol{d}\|$ is the normalised version,
- $\boldsymbol{\omega}_p$ is the direction of propagation of the photon,
- F_t is the Fresnel term for transmission. If the angle between $\boldsymbol{d}$ and the normal $\boldsymbol{n}$ is larger than $\sin^{-1} \frac{1}{\eta}$, then the photon is fully reflected at the material interface and does not contribute to this specific sample point.

We sample the BSSRDF for the incoming directions: $\theta_i \in \{0, 15, 30, 45, 60, 70, 80, 88\}$ at sample points defined by: $r \in \{0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 8, 10\}$. Directions θ_s are sampled regularly every 15 degrees. We store outgoing directions θ_o and φ_o in buckets of width 5 degrees. We used the same sampling strategy as [DLR+09], for easier comparison and validation.

The whole simulation uses roughly 350 lines of code, and we provide it as supplemental material for researchers wishing to reproduce our results[2]. We checked that we reproduce the results of [DLR+09], even though we use a different method for sampling outgoing directions: we used regular samples in θ_o and φ_o while they used directions uniformly distributed on the sphere using the Healpix format.

3.3 Correlation between two-scatter events and multiple scatter events

In this paper, we only focus on multiple scattering events. Single scattering events are more complex and exhibit subtle spatial variations [WZHB09].

Among multiply scattered photons, we separate between photons scattered exactly twice and the others. We compute the outgoing light leaving the material after exactly

2. See the IEEE digital library for this paper.

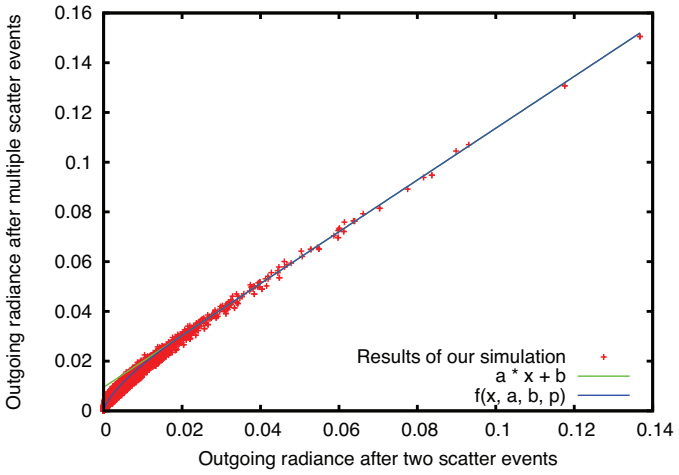


Figure 2: Relationship between radiance leaving the material after two scatter events (x) and multiple scatter events (y). The two quantities are strongly correlated, with $r = 0.998$ ($\theta_i = 0$, $\alpha = 0.95$, $g = 0.3$, $\eta = 1.3$).

two scatter events, along with the light leaving the material after multiple scatter events. Figure 2 shows the relationship between these two quantities, measured at multiple sample points and directions on the surface. Looking at the curve, you can identify two areas: Where there is a strong double scattering ($L > 0.02$), there is also a strong correlation between double and multiple scattering. When double scattering is negligible ($L < 0.02$), the relationship is not linear, although double scattering is still a good predictor of single scattering.

Figure 2 corresponds to a single material: $\eta = 1.3, \alpha = 0.95, g = 0.3$, and a single input direction, $\theta_i = 0$. We conducted an extensive study for a complete range of materials, and found that this strong correlation always exists, except for materials with a strong backward scattering ($g < -0.5$). Note that Figure 2 corresponds to a material with a very high albedo, giving a strong importance to multiple scattering.

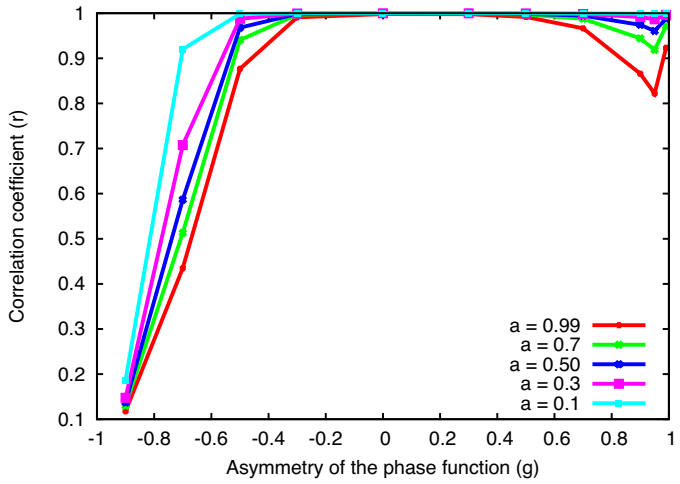
In our study, we tried all possible combinations of material parameters and input directions. For each material, we compute the correlation coefficient between outgoing light due to multiple scatter events and the light due to only two scatter events. For two distributions X and Y, the correlation coefficient r_{XY} is defined as:

$$r_{XY} = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

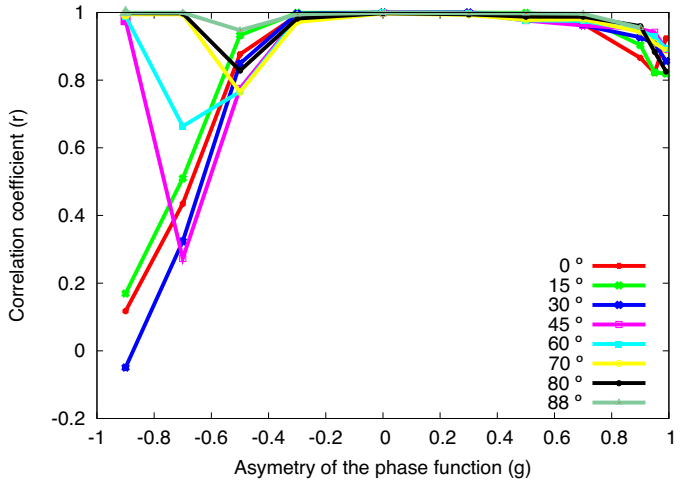
r_{XY} is equal to 1 for perfectly correlated distributions, and 0 if the distributions are independent from each other. We found that the relationship holds for all materials with $g > -0.5$, for all input directions. Thus light leaving after two scatter events is a strong predictor of multiply scattered light, except for materials with strong backward scattering properties.

Figure 3 presents our results. It displays the variation of the correlation coefficient as a function of the input parameters of the simulation. A coefficient $r > 0.95$ corresponds to a very strong correlation; $r > 0.8$ corresponds to a strong correlation.

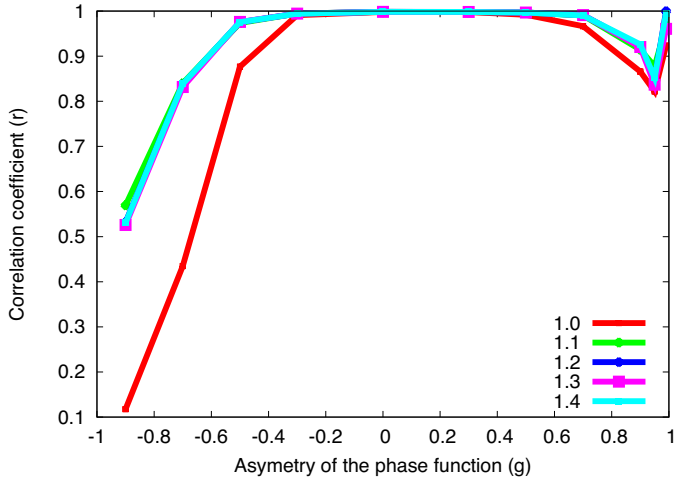
Albedo: Figure 3(a) shows the variation of the correlation coefficient r as a function of g for different values of the albedo α. At each scatter event, the light is multiplied by α. Reducing the albedo reduces the amount of light being multiply scattered,



(a) Albedo α ($\eta = 1, \theta_i = 0$)



(b) Input angle θ_i ($\alpha = 0.99, \eta = 1$)



(c) Index of refraction η ($\alpha = 0.99, \theta_i = 0$)

Figure 3: Evolution of the correlation coefficient r between second-order scatter events and multiple scatter events, depending on the simulation parameters. Observe that we consistently observe a strong correlation for materials with $g > -0.5$.

increasing the importance of two-event scatterings among multiple scattering, and increasing the correlation coefficient. In all figures except this one, we used a large value for the albedo ($\alpha = 0.99$): the correlations we find can only get better with lower values of the albedo.

For strongly backward scattering materials ($g < -0.5$), there is no correlation between second-order scattering and multiple scattering.

Incoming direction: Figure 3(b) shows the variation of the correlation coefficient r as a function of g for different values of the input direction θ_i. When there is a strong correlation between second-scatter and multiple-scatter, we see that this correlation remains the same, independently of the input direction: for $g > -0.5$, the curves follow similar patterns, even though there are slight variations. For strongly backward scattering materials, we see more variations in the correlation coefficient (there is more correlation at grazing angles), but as we are looking for a model that is independent from input direction this information is not very useful to us.

More importantly, the correlation we find is also independent on θ_i: for $g = 0$, $\alpha = 0.99$, $\eta = 1$ (illustrated on Figure 2) we consistently get $y \approx 1.04x + 0.02$, for all θ_i. This key property means that we can replace the computation of multiple scattering by a computation of two-event scattering, and then deduce multiple scattering by interpolating. Note that in this case (high albedo, isotropic scattering), two-event scattering amounts for 96 % of multiple scattering.

Index of refraction: Figure 3(c) shows the variation of the correlation coefficient r as a function of g for different values of the index of refraction η. As you can see, η has little influence on the quality of the correlation. The main difference is between $\eta = 1$ (no refraction at the interface) and $\eta > 1$. Materials with a index of refraction larger than 1 can experience total internal reflection: light reaching the surface above a certain angle does not exit and is reflected towards the material. This effect reduces both two-events scattering and multiple scattering, but apparently increases the correlation between the two.

In all figures except this one, we placed ourselves in the worst case scenario: $\eta = 1$. Thus changing the index of refraction can only increase the correlations we found.

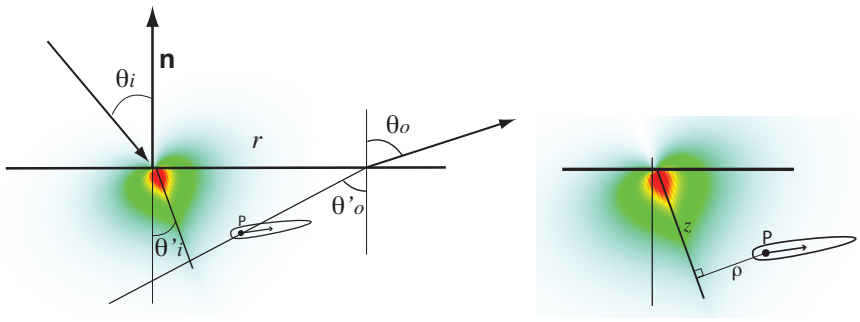
3.4 Conclusion and discussion

We have found a strong correlation between light exiting the material after two scattering events and multiply scattered light, for a large range of materials (forward scattering and moderately backward scattering, $g > -0.5$). We found no visible correlation for strongly backward scattering materials ($g < -0.5$). The strong correlation holds for all values of the albedo, incoming direction and index of refraction.

We can use this correlation to speed-up the computation of multiple scattering in translucent materials: we compute two-event scattering and extrapolate to extract multiple scattering. The correlation coefficients depend on the actual material parameters (g, α and η) but can be precomputed. Figure 7 illustrates the results of this extrapolation. Where double scattering is low, we cannot use the affine correlation $y = ax + b$, as it would overestimate multiple scattering for low values of double scattering. We extended the affine correlation in a C^1 way so that its value is null at the origin (see Figure 2):

$$f(x, a, b, p) = \begin{cases} ax + b & \text{if } x > x_0 \\ \frac{b}{1-p}\left(\frac{x}{x_0}\right)^p & \text{if } x \leq x_0 \text{ where } x_0 = \frac{pb}{a(1-p)} \end{cases}$$

We obtain the a, b and p parameters by fitting the computed values. The fact that multiply-scattered light is strongly correlated with two-event scattering explains a key re-



(a) Using the density map

(b) Cylindrical coordinates (ρ, z), centered on the refracted ray

Figure 4: We compute the density of two scatter events around the scattered ray inside the material, along with average direction. At rendering time, we find the contribution at sample point P using the density, average direction and phase function.

sult of Donner *et al.* [DLR+09]: they found that for most materials, lobes were aligned with the scattering plane (defined by the entry point and direction and the exit point). This is obviously true for light scattered twice. Since multiply-scattered light is correlated to that, we naturally get the same result.

We expect this result can be used with any light simulation algorithm such as path tracing, Monte-Carlo ray-tracing or Bi-Directional Path Tracing to speed up the computation of sub-surface scattering. In the next section, we provide a specific algorithm for low-memory computation of two-event scattering.

4 Fast computation of two-scatter events

In this section, we present an algorithm for fast computation of double-scattering events in translucent materials. When the light ray enters the material, it changes direction with refraction, but keeps propagating in a straight line. All the first scattering events take place on this line. As a consequence the distribution of second-scattering events has rotational symmetry around this line. We exploit this property to store photon density in a compact way.

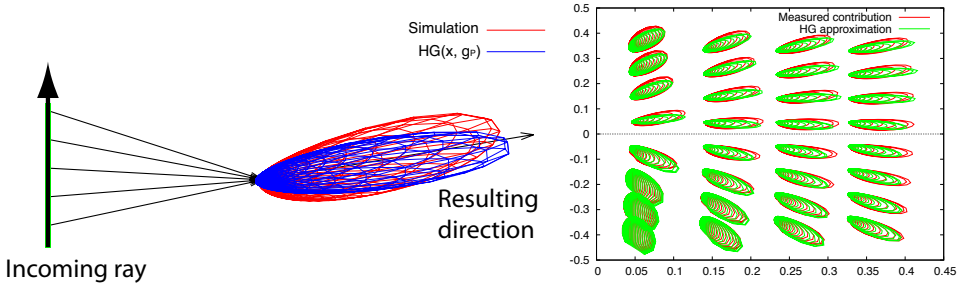
We store the photon density in the medium, using the refracted incoming ray as the main axis for our frame. Since the photon density has rotational symmetry around this axis, we simply have to store a 2D function of ρ and z.

In the rendering step, for each outgoing ray, we compute the refracted ray inside the material and extract the contribution from the second scatter event on this ray (see Figure 4).

4.1 Precomputation: density and direction

Figure an incoming ray hitting the material, at an angle θ_i with the normal. This ray is refracted when it hits the surface. The new ray makes an angle θ'_i with the surface normal, with θ'_i defined by Snell's law: $\eta \sin \theta'_i = \sin \theta_i$. We express the incoming ray after refraction as R.

All first scatter events take place on R. As a consequence, the probability distribution of second scatter events has rotational symmetry around R (if we allow second scatter



(a) Summing up all contributions at a given point

(b) Lobes, measured and approximated, for sample points

Figure 5: We approximate the contributions at each sample point with a single Henyey-Greenstein lobe, with direction $\boldsymbol{v}$ and anisotropy g.

events to take place above the surface of the material). We introduce cylindrical coordinates based on R: (ρ, z) (see Figure 4). We compute three quantities:

- the probability density of second scatter events, d,
- the average direction for all photons scattered twice reaching that point $\boldsymbol{v}$, stored as an angle in 2D, θ.
- the anisotropy of the directions of photons reaching this point, expressed as the g parameter of a Henyey-Greenstein phase function, and computed using equation 3.

We store all second scatter events, even those happening outside of the medium, and store them before multiplication by the albedo. Thus the quantities we compute and store only depend on the anisotropy of the phase function g, not on the incoming direction θ_i, the index of refraction, or the albedo.

We do our precomputation in 2D-space for better efficiency. The density in 3D is connected to the density we computed in 2D by:

$$d_{3D}(\rho, z) = \frac{1}{2\pi\rho} d_{2D}(\rho, z) \tag{7}$$

After precomputation, we store these functions into 2D textures. Using a texture resolution of 128×256, the cost for a three-channels, 4 byte-per-channel texture is 512 KB. The overall cost for storing 20 textures sampling g regularly from -0.9 to 0.95 is 7.5 MB.

To compute the main direction $\boldsymbol{v}$ and anisotropy, at each sample point, we compute the outgoing radiance in all directions, taking into account the phase function and exponential falloff at both scatter events. We sum the contributions for all primary scatter events, resulting in an anisotropic lobe (see Figure 5). We approximate this anisotropic lobe with a Henyey-Greenstein phase function, using equation 3, and store its main direction and value.

4.2 Sampling at runtime

At runtime, we have a sample point x_o and an outgoing direction ω_o and we need the outgoing light at this point. We takes ω'_o the refracted version of the outgoing ray (the ray that is such that $\eta \sin\theta'_o = \sin\theta_o$). We call S the refracted outgoing ray: (x_o, ω'_o).

If we consider a single point P along S, we obtain the contribution of double scattering on this point P by:

- express P in the frame of reference of the incoming ray R and divide coordinates by the mean-free-path.
- extract the density d_P, main direction $\boldsymbol{v}_P$ and anisotropy g_P at point P by bi-linear interpolation from the pre-computed values,
- compute the phase function $p\left(\boldsymbol{v}_P \cdot \boldsymbol{\omega}'_o, g_P\right)$,
- and finally multiplying by the density, albedo, Fresnel term and density volume element at this sampling point, $\mathrm{d}V$:

$$L_P = \sigma_s^2 d_P p\left(\boldsymbol{v}_P \cdot \boldsymbol{\omega}'_o, g_P\right) F_t(\boldsymbol{\omega}'_o)\,\mathrm{d}V \tag{8}$$

Depending on our application, we either want to use the contribution of a single point P (for example to connect light and eye paths in bi-directional path-tracing) or integrate the contributions of all points P along the ray S. For the latter, we sample at points P along the ray using the exponential falloff as importance function. We sample regularly for $\xi \in [0, 1[$ and take $P = x_o - \log(1-\xi)\boldsymbol{\omega}'_o$. This reduces the number of sample points while taking into account the exponential falloff with the distance.

4.3 Direct BSSRDF lobes comparison

Figure 6 shows a comparison between the lobes for two scatter events generated by our approximation and Monte-Carlo simulation, for different values of the material parameters. Notice how our approximation fits all the features of the lobes computed using Monte-Carlo approximation, both for size and shape. We are even able to capture the secondary lobe which happens for slanted incoming lighting (see second row).

4.4 Comparison with full Monte-Carlo simulation in a simple case

Figure 7 shows a side-by-side comparison between our algorithm and a reference, computed using Monte-Carlo simulation. We chose a simple setting so that all illumination effects would be visible: a collimated beam of light hitting a half-space with a planar surface, filled with a blue translucent material: $\sigma_s = (0.07, 0.53, 0.52)$, $\sigma_a = (0.93, 0.47, 0.48)$, g = 0. We used a refraction index of $\eta = 1.0$ to visualize only scattering effects.

The two pictures on the left display only single and double scattering. You can see that our algorithm provides a good approximation of double scattering. The main problem is that we underestimated the light propagated along the direction of the beam (visible on the right edge of the picture). We attribute this to under-sampling for small values of r. Even for isotropic material, most of the interesting events take place close to the original beam. Instead of regular sampling, we could use more samples for small values of r.

The two pictures on the right show all scattering events. We computed multiple scattering by extrapolation from double scattering in our algorithm, and compare it with the reference solution using Monte-Carlo. Our algorithm exhibits the right behaviour in all directions. We tend to underestimate illumination in the direction of the beam (this is a consequence of underestimating double scattering), and to overestimate it in areas at the periphery of the halo.

4.5 Integration in a ray-tracer

We ported our algorithm in the Mitsuba renderer [Jak10], as an extension of its `subsurface` class, which uses Jensen and Buhler [JB02] method. Our implementation is an extension of [JB02]: first, we precompute incoming light at randomly distributed sample points and directions on the surface of the object. In a second pass, we integrate the contributions from these directional samples. All rendered images in this paper were generated using this implementation. The main difference with [JB02] is that we need to store the

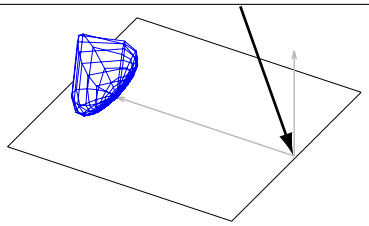
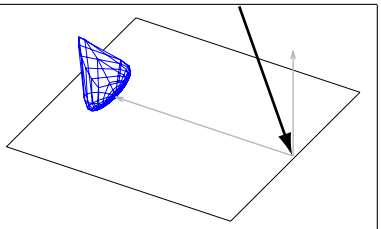
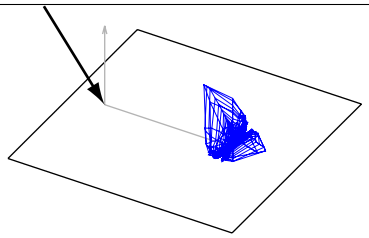
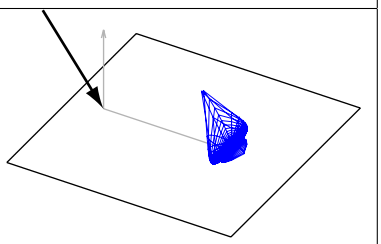
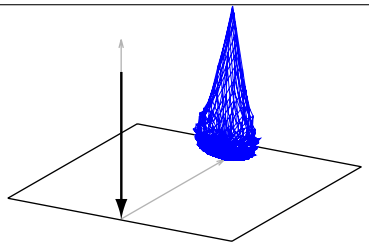
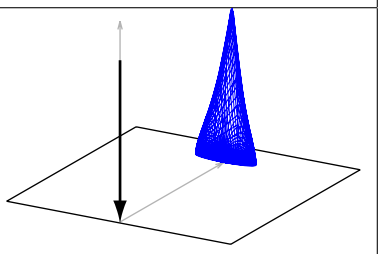
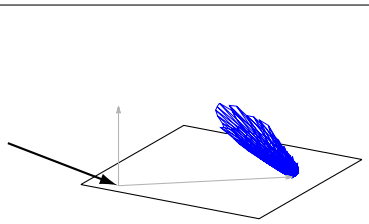
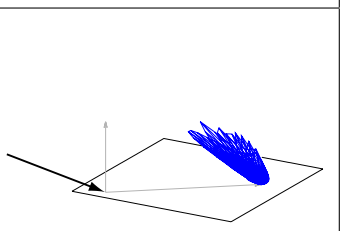
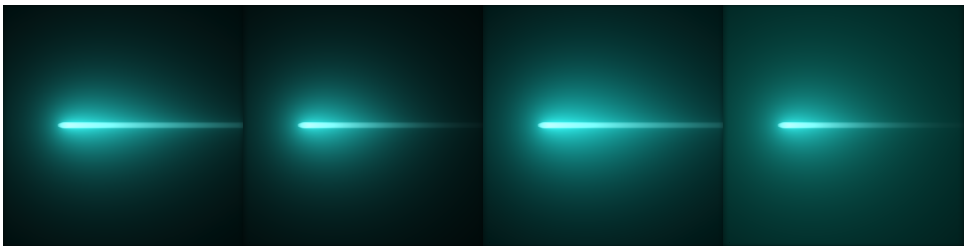
Parameters	Ground Truth	Our algorithm
$\eta = 1.0$ $\theta_i = 60^o$ $g = 0.5$ $\alpha = 0.99$ $r = 0.8$ $\theta_s = \pi$		
$\eta = 1.0$ $\theta_i = 60^o$ $g = 0.5$ $\alpha = 0.99$ $r = 0.4$ $\theta_s = 0$		
$\eta = 1.3$ $\theta_i = 0^o$ $g = -0.3$ $\alpha = 0.99$ $r = 0.8$ $\theta_s = \pi/2$		
$\eta = 1.2$ $\theta_i = 70^o$ $g = 0.7$ $\alpha = 0.99$ $r = 0.8$ $\theta_s = \pi/3$		

Figure 6: Comparison between two-scattering lobes computed using our approximation (left) and Monte-Carlo (1.2 million photons). Both columns are using the same scaling. Notice how we pick the right direction and shape for the lobes, including the presence of a secondary lobe on the second row.



(a) Reference, single and double scattering

(b) Our algorithm, single and double scattering

(c) Reference, single and multiple scattering

(d) Our algorithm, single and multiple scattering extrapolated from double scattering

Figure 7: A collimated beam, making an angle of $\pi/3$ with the surface normal, hitting the surface of a half-space containing a blue translucent material. Comparison between values computed using our algorithm and full Monte-Carlo simulation. Material: $\sigma_t = 1$, $\alpha = (0.07, 0.53, 0.52)$, $\eta = 1$, $g = 0$.

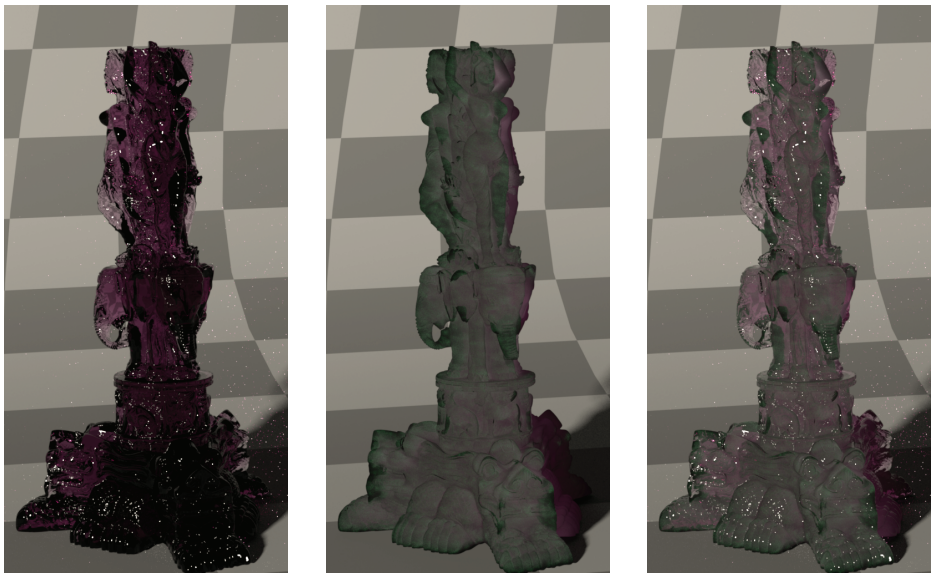


Figure 8: Compositing the different components of the model. From left to right: all effects except scattering, all scattering effects (single, double and multiple), and the full model (reflections, refraction and scattering). Material: $\sigma_t = (0.44, 0.88, 0.55)$, $\alpha = 0.6$, $\eta = 1.3$, $g = 0.5$.

incoming radiance in both space and direction. This significantly increases the number of samples we need to store at runtime, depending on material density and anisotropy of the phase function.

Figure 1 displays a comparison between our method (center column), the classical dipole approximation and a reference solution. For the reference solution, we used bi-directional path-tracing and a large number of samples per pixel (4,000). All pictures were computed using the same material definition, inside the same renderer (Mitsuba). The computations were spread over three networked computers with Intel core 2 processors, for a total of 16 cores (two machines with 4 cores, one machine with 8 cores). We used 32 samples per pixel for the dipole approximation, for more accurate shadows and anti-aliasing, and 16 for our implementation.

The top row shows multiple scattering effects *only*: the result of the dipole approximation (left), extrapolating from double scattering for our method (center) and multiple scattering effects for the reference solution. To present only multiple scattering effects for the reference solution, we used the following method: first, we computed a full reference solution, with all effects included. Then, we subtracted from it a picture with only reflection, refraction and single scattering, computed by restricting the number of events in bi-directional path tracing.

The bottom row shows the picture with all lighting effects: reflection, refraction, translucency and single scattering combined with multiple scattering (for bi-directional path-tracing, it corresponds to the picture computed without any tweaks).

The dipole approximation is very fast (16 seconds), but differs in terms of color. Our implementation, using precomputed double scattering effects and extrapolating multiple scattering, provides the right color, while being 20 times faster than the reference solution, and only 6 times slower than the dipole approximation

Figure 8 displays the relative contributions of all the elements used in rendering translucent materials: specular reflections on the surface, transmittance through the material, both of these combined, then multiple scattering (extrapolated from double scattering) and the full model. The effects of scattering are clearly visible on the central part of the statue.

Figure 9 shows the effects of varying material density, while keeping other material parameters constant. This is equivalent to changing the scale of the object compared to the mean-free-path. We show separately the effects of reflection and refraction (left column), multiple scattering extrapolated from double scattering (center column) and all of them combined together (right column). As the object gets larger (top row) or smaller (bottom row), scattering effects are less visible, compared to reflection and refraction effects. Scattering effects are more visible if the object has the right size compared to the mean-free-path, not too large nor too small.

5 Acknowledgments

The Armadillo, Asian dragon and Thai statuette models appearing in the figures of this paper are kindly provided by the Stanford Computer Graphics Laboratory. All pictures were ray-traced using the Mitsuba renderer [Jak10], with our custom plug-in for double scattering effects.

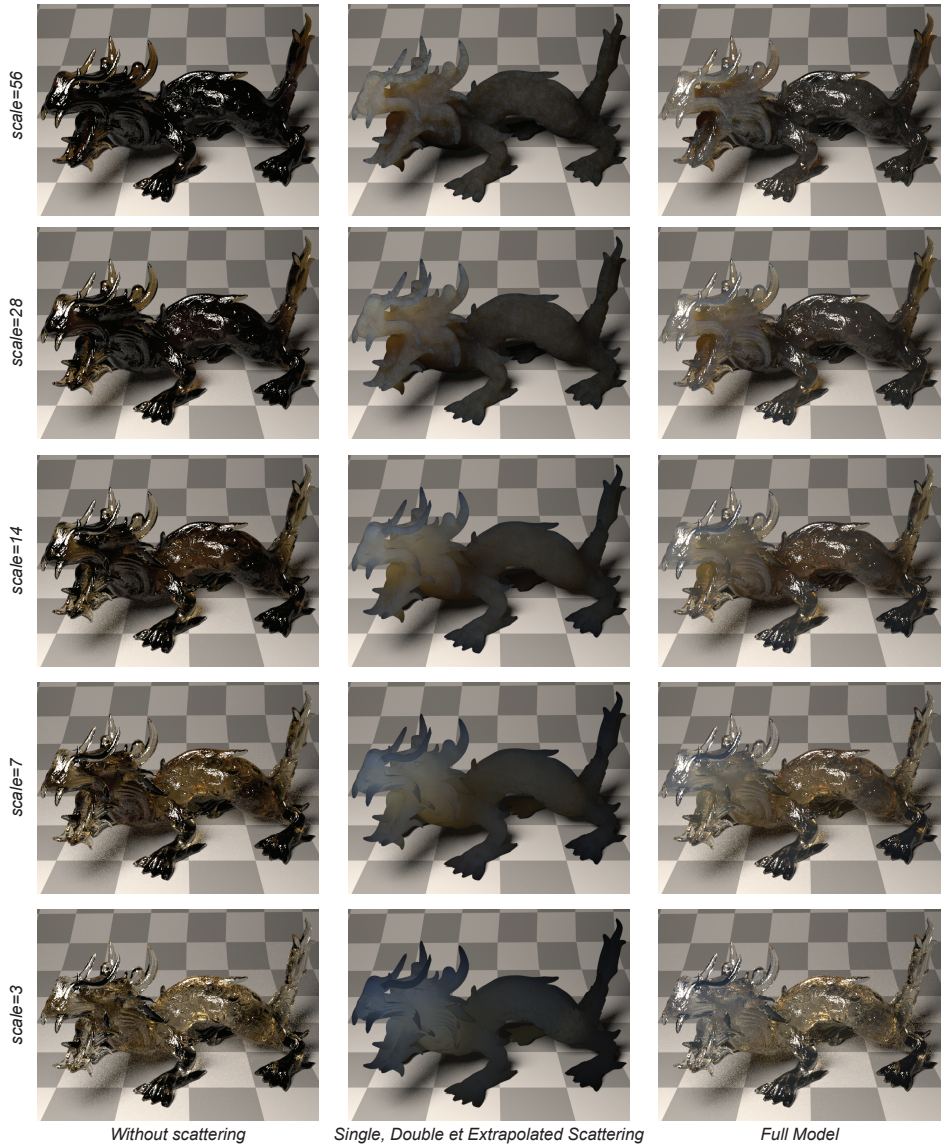


Figure 9: The effects of varying material density. Left column: only reflected and transmitted light. Central column: scattered light (single scattering plus multiple scattering, extrapolated from double scattering). Right column: complete lighting. Material: $\sigma_s = (0.2, 0.3, 0.6)$, $\alpha = 0.9$, $\eta = 1.3$, $g = 0.2$.

6 Conclusion

Our first contribution is a study of sub-surface scattering effects in translucent materials. Except for materials with strong backward scattering ($g < -0.5$), our experiments show a strong correlation between multiple scattering and double scattering. Any light simulation algorithm can exploit this property to speed up the computations of multiple scattering in sub-surface scattering, by computing only the double scattering and extrapolating.

What we have found is a an affine correlation: it does not mean that multiple scattering does not exist after the second event, it means that, on average, its distribution can be predicted by the results of the second event. Our study was only conducted on surface events; we haven't studied the distribution of light *inside* translucent materials. This would require a separate study.

Our second contribution is an approximate model to compute double-scattering effects in an efficient way. We store the average material response, independently from the incoming direction. We use these precomputed values to get the response from the second scatter event, and extrapolate to get multiple scattering. We integrated this model inside a sub-surface scattering plug-in, but this model could be combined with any illumination simulation algorithm. Our model is an approximation, but provides accurate colors and fair light distribution, while being very effective in terms of memory and CPU.

References

[DI11] Eugene D'Eon and Geoffrey Irving, *A quantized-diffusion model for rendering translucent materials*, ACM Trans. Graph. **30** (2011), no. 4, 56:1–56:14.

[DLR+09] Craig Donner, Jason Lawrence, Ravi Ramamoorthi, Toshiya Hachisuka, Henrik Wann Jensen, and Shree Nayar, *An empirical bssrdf model*, ACM Trans. Graph. **28** (2009), no. 3, 30:1–30:10.

[HG41] Louis G. Henyey and J. L. Greenstein, *Diffuse radiation in the galaxy*, Astrophysical Journal **93** (1941), 70–83.

[HK93] Pat Hanrahan and Wolfgang Krueger, *Reflection from layered surfaces due to subsurface scattering*, 20th annual conference on Computer graphics and interactive techniques, SIGGRAPH '93, 1993, pp. 165–174.

[Jak10] Wenzel Jakob, *Mitsuba renderer*, 2010, http://www.mitsuba-renderer.org.

[JB02] Henrik Wann Jensen and Juan Buhler, *A rapid hierarchical rendering technique for translucent materials*, ACM Trans. Graph. **21** (2002), no. 3, 576–581.

[Jen01] Henrik Wann Jensen, *Realistic image synthesis using photon mapping*, A. K. Peters, Ltd., Natick, MA, USA, 2001.

[JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan, *A practical model for subsurface light transport*, 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01, 2001, pp. 511–518.

[Kaj86] James T. Kajiya, *The rendering equation*, SIGGRAPH Computer Graphics **20** (1986), no. 4, 143–150.

[WZHB09] Bruce Walter, Shuang Zhao, Nicolas Holzschuch, and Kavita Bala, *Single scattering in refractive media with triangle mesh boundaries*, ACM Trans. Graph. **28** (2009), no. 3, 92:1–92:8.